

Learning optimally diverse rankings over large document collections

(Full version: May 2010)

Aleksandrs Slivkins

Microsoft Research, 1065 La Avenida, Mountain View CA, USA

SLIVKINS@MICROSOFT.COM

Filip Radlinski

Microsoft Research, 7 J.J. Thomson Ave., Cambridge UK

FILIPRAD@MICROSOFT.COM

Sreenivas Gollapudi

Microsoft Research, 1065 La Avenida, Mountain View CA, USA

SREENIG@MICROSOFT.COM

Abstract

Most learning to rank research has assumed that the utility of different documents is independent, which results in learned ranking functions that return redundant results. The few approaches that avoid this have rather unsatisfyingly lacked theoretical foundations, or do not scale. We present a learning-to-rank formulation that optimizes the fraction of satisfied users, with a scalable algorithm that explicitly takes document similarity and ranking context into account. We present theoretical justifications for this approach, as well as a near-optimal algorithm. Our evaluation adds optimizations that improve empirical performance, and shows that our algorithms learn orders of magnitude more quickly than previous approaches.

1. Introduction

Identifying the most relevant results to a query is a central problem in web search, hence learning ranking functions has received a lot of attention (e.g., [Burges et al., 2005](#); [Chu and Ghahramani, 2005](#); [Taylor et al., 2008](#)). One increasingly important goal is to learn from user interactions with search engines, such as clicks. We address the task of learning a ranking function that minimizes the likelihood of query abandonment (i.e. no click). This objective is particularly interesting as query abandonment is a major challenge in today’s search engines, and is also sensitive to the diversity and redundancy among documents presented.

We consider the Multi-Armed Bandit (MAB) setting

(e.g. [Cesa-Bianchi and Lugosi, 2006](#)), which captures many online learning problems wherein an algorithm chooses sequentially among a fixed set of alternatives (“arms” or “strategies”). MAB algorithms are ideal for online settings with exploration/exploitation tradeoffs. While most MAB literature corresponds to learning a single best alternative (*single-slot* MAB), MAB algorithms can also be extended to *multiple slots*, e.g. to learn a ranking of documents that minimizes query abandonment ([Radlinski et al., 2008](#); [Streeter and Golovin, 2009](#)). However, most MAB algorithms are impractical at web scales.

Prior work on MAB algorithms has considered exploiting structure in the strategy space to improve convergence rates. One particular approach, articulated by [Kleinberg et al. \(2008b\)](#) is well suited to our scenario: when the strategies (in our case, documents) form a metric space and the payoff function satisfies a Lipschitz condition with respect to the metric. The metric space allows the algorithm to make inferences about similar documents without exploring them. Further, they propose a “zooming algorithm” that learns to adaptively refine explored regions of the strategy space where there is likelihood of higher payoff, and provide strong provable guarantees about its performance.

In web search, a metric space directly models similarity between documents.¹ Further, one can use additional signals. A search user typically scans results top down, and clicks on more relevant documents. One can therefore infer the *context* in which a click happened: the skipped documents at higher ranks. To fully exploit the context we factor in both *conditional clickthrough rates* and *correlated clicks*. The former conditions on the event that the user skipped a set of documents (as suggested by [Chen and Karger, 2006](#)), and the latter refers to the probability that two documents are both

Full version of the paper in *27th International Conference on Machine Learning (ICML)*, 2010.

¹In fact, most offline learning-to-rank approaches also rely on similarity between documents, at least implicitly.

relevant (or irrelevant) to a given user.

Our contributions. This paper initiates the study of online learning-to-rank in metric spaces. We propose a simple learning model that explicitly considers correlation of clicks and similarity between documents, and admits efficient bandit algorithms that, unlike those in prior work on bandit-based learning-to-rank, scale to large document collections. We study this model both theoretically and empirically. First, we validate the expressiveness of our model by providing an explicit construction for a wide family of plausible user distributions which provably fit the model. Second, we design several algorithms for our model, joining and extending ideas from “ranked bandits” (Radlinski et al., 2008), bandits on metric spaces (Kleinberg et al., 2008b) and contextual bandits (Slivkins, 2009).² Third, we provide provable scalability guarantees. Finally, we empirically study their performance using the above-mentioned construction with realistic parameters.

In a more abstract view, we tackle the problem of using side information on document similarity in the online learning-to-rank setting. We focus on the case of “ideally clean” similarity data, with a two-pronged high-level question: how to model such data, and how to use it algorithmically. We believe that studying the “clean” case is useful (and perhaps necessary) to inform and guide the corresponding data-driven work.

Outline. We define the model in Section 2, validate its expressiveness in Section 3, design algorithms in Section 4, prove scalability guarantees in Section 5, and discuss simulations in Section 6. It is worth noting that much of the theoretical (provable) contribution concerns setting up various aspects of the model in Section 2.2, Section 3, and Section 4.4. To keep the flow of the paper, the lengthy proofs from Section 3 are presented in Section 7.

2. Multi-slot bandits in metric spaces

Let us introduce and motivate the online learning-to-rank problem that we study in this paper.

2.1. Problem formalization

Learning problem. Following (Radlinski et al., 2008), we are interested in learning an optimally diverse ranking of documents for a given query. We model it as a *multi-slot* MAB problem as follows. Let

²For more work on (contextual) bandits on metric spaces, see (Agrawal, 1995; Kleinberg, 2004; Pandey et al., 2007; Auer et al., 2007; Hazan and Megiddo, 2007; Bubeck et al., 2008; Kleinberg and Slivkins, 2010).

X be a set of documents (“arms”). Each ‘user’ is represented by a binary *relevance vector*: a function $\pi : X \rightarrow \{0, 1\}$. A document $x \in X$ is called “relevant” to the user if and only if $\pi(x) = 1$. Let \mathcal{F}_X be the set of all possible relevance vectors. Users come from a distribution \mathcal{P} on \mathcal{F}_X that is fixed but not revealed to an algorithm.³ In each round, the following happens: a user arrives, sampled independently from \mathcal{P} ; an algorithm outputs a list of k results; the user scans the results top-down, and clicks on the first relevant one. The goal is to maximize the expected fraction of users who click on a result. Note that in contrast with prior work on diversifying existing rankings (e.g. Carbonell and Goldstein, 1998), our aim is to directly learn a diverse ranking.

Click probability. The *pointwise mean* of \mathcal{P} is a function $\mu : X \rightarrow [0, 1]$ such that $\mu(x) \triangleq \mathbb{E}_{\pi \sim \mathcal{P}}[\pi(x)]$. Thus, $\mu(x)$ is the click probability for document x if it appears in the top slot. Each slot $i > 1$ is examined by the user only in the event that all documents in the higher slots are not clicked, so the relevant click probabilities for this slot are conditional on this event. Formally, fix a subset of documents $S \subset X$ and let $Z_S \triangleq \{\pi(\cdot) = 0 \text{ on } S\}$ be the event that all documents in S are not relevant to the user. Let $(\mathcal{P}|Z_S)$ be the distribution of users obtained by conditioning \mathcal{P} on this event, and let $\mu(\cdot|Z_S)$ be its pointwise mean.

Metric spaces. Throughout the paper, let (X, \mathcal{D}) be a *metric space*.⁴ A function $\nu : X \rightarrow \mathbb{R}$ is said to be *Lipschitz-continuous* (*L-continuous*) with respect to (w.r.t.) (X, \mathcal{D}) if

$$|\nu(x) - \nu(y)| \leq \mathcal{D}(x, y) \quad \text{for all } x, y \in X. \quad (1)$$

A user distribution \mathcal{P} is called *L-continuous* with respect to (X, \mathcal{D}) if its pointwise mean μ is L-continuous w.r.t. (X, \mathcal{D}) .

Document similarity. To allow us to incorporate information about similarity between documents, we start with the model proposed by Kleinberg et al. (2008b), called *Lipschitz MAB*: an algorithm is given a metric space (X, \mathcal{D}) w.r.t. which the pointwise mean μ is L-continuous.⁵ This model suffices for learning the document at the top position ($k = 1$).

However, for lower ranked documents this model is not

³Note that this also models users for whom documents are probabilistically relevant (Radlinski et al., 2008)

⁴I.e., X is a set and \mathcal{D} is a non-negative symmetric function on $X \times X$ such that (i) $\mathcal{D}(x, y) = 0 \iff x = y$, and (ii) $\mathcal{D}(x, y) + \mathcal{D}(y, z) \geq \mathcal{D}(x, z)$ (triangle inequality).

⁵One only needs to assume that similarity between any two documents x, y is summarized by a number $\delta_{x,y}$ such that $|\mu(x) - \mu(y)| \leq \delta_{x,y}$. Then one obtains a metric space by taking the shortest paths closure.

sufficiently informative since the relevant click probabilities $\mu(\cdot|Z_S)$ are conditional. We will assume a stronger property called *conditional L-continuity*:

Definition 2.1. \mathcal{P} is called *conditionally L-continuous w.r.t. (X, \mathcal{D})* if the conditional pointwise mean $\mu(\cdot|Z_S)$ is L-continuous for all $S \subset X$.

Now, a document x in slot $i > 1$ is examined only if event Z_S happens, where S is the set of documents in the higher slots. x has a conditional click probability $\mu(x|Z_S)$. The function $\mu(\cdot|Z_S)$ satisfies the Lipschitz condition (1), which will allow us to use the machinery from MAB problems on metric spaces.

Formally, we define the *k-slot Lipschitz MAB problem*, an instance of which consists of a triple $(X, \mathcal{D}, \mathcal{P})$, where (X, \mathcal{D}) is a metric space that is known to an algorithm, and \mathcal{P} is a latent user distribution which is conditionally L-continuous w.r.t. (X, \mathcal{D}) .

This formulation includes the “metric-less” *k-slot MAB problem* from (Radlinski et al., 2008) as a special case in which the metric space is “trivial”: all distances are 1. In particular, all algorithms in Section 4 are well-defined for the “metric-less” version.

2.2. Correlated relevance

An alternative notion of document similarity focuses on *correlated relevance*: correlation between the relevance of two documents to a given user. We express “similarity” by bounding the probability of the “dis-correlation event” $\{\pi(x) \neq \pi(y)\}$. Specifically, we consider *conditional L-correlation*, defined as follows:

Definition 2.2. Call \mathcal{P} L-correlated w.r.t. (X, \mathcal{D}) if

$$\Pr_{\pi \sim \mathcal{P}} [\pi(x) \neq \pi(y)] \leq \mathcal{D}(x, y) \quad \forall x, y \in X. \quad (2)$$

Call \mathcal{P} conditionally L-correlated w.r.t. (X, \mathcal{D}) if (2) holds conditional on Z_S for any $S \subset X$, i.e.

$$\Pr_{\pi \sim (\mathcal{P}|Z_S)} [\pi(x) \neq \pi(y)] \leq \mathcal{D}(x, y) \quad \forall x, y \in X, S \subset X.$$

It is easy to see that conditional L-correlation implies conditional L-continuity. In fact, we show that the two notions are essentially equivalent. Namely, we prove that conditional L-continuity w.r.t. (X, \mathcal{D}) implies conditional L-correlation w.r.t. $(X, 2\mathcal{D})$.

Lemma 2.3. Consider an instance $(X, \mathcal{D}, \mathcal{P})$ of the *k-slot Lipschitz MAB problem*. Then the user distribution \mathcal{P} is conditionally L-correlated w.r.t. $(X, 2\mathcal{D})$.

Proof. Fix documents $x, y \in X$ and a subset $S \subset X$. For brevity, write “ $x = 1$ ” to mean “ $\pi(x) = 1$ ”, etc.

We claim that

$$\Pr[x = 1 \wedge y = 0 | Z_S] \leq \mathcal{D}(x, y). \quad (3)$$

Indeed, consider the event $Z = Z_{S+\{y\}}$. Applying the Bayes theorem to $(\mathcal{P}|Z_S)$, we obtain that

$$\begin{aligned} \mu(x|Z) &= \Pr[x = 1 | \{y = 0\} \wedge Z_S] \\ &= \frac{\Pr[x = 1 \wedge y = 0 | Z_S]}{\Pr[y = 0 | Z_S]}. \end{aligned} \quad (4)$$

On the other hand, since $\mu(y|Z) = 0$, by conditional L-continuity it holds that

$$\mu(x|Z) = |\mu(x|Z) - \mu(y|Z)| \leq \mathcal{D}(x, y), \quad (5)$$

so the claim (3) follows from (4) and (5).

Likewise, $\Pr[x = 0 \wedge y = 1 | Z_S] \leq \mathcal{D}(x, y)$. Since

$$\{\pi(x) \neq \pi(y)\} = \{x = 1 \wedge y = 0\} \cup \{x = 0 \wedge y = 1\},$$

it follows that $\Pr[\pi(x) \neq \pi(y) | Z_S] \leq 2\mathcal{D}(x, y)$. \square

2.3. Metric space: a running example

Web documents are often classified into hierarchies, where closer pairs are more similar.⁶ For evaluation, we assume the documents X fall in such a tree, with each document $x \in X$ a leaf in the tree. On this tree, we consider a very natural metric: the *ϵ -exponential tree metric*: the distance between any two tree nodes u, v is exponential in the height of their least common ancestor, with base $\epsilon \in (0, 1)$:

$$\mathcal{D}(u, v) = c \times \epsilon^{\text{height}(\text{LCA}(u, v))}, \quad (6)$$

for some constant c . However, our algorithms and analyses extend to arbitrary metric spaces.

3. Expressiveness of the model

Our approach relies on the conditional L-continuity (equivalently, conditional L-correlation) of the user distribution. How “expressive” is this assumption, i.e. how rich and “interesting” is the collection of problem instances that satisfy it? While the unconditional L-continuity assumption is usually considered reasonable from the expressiveness point of view, even the unconditional L-correlation (let alone the conditional one) is a very non-trivial property about correlated relevance, and thus potentially problematic. A related concern is how to generate a suitable collection of problem instances for simulation experiments.

In this section we address both concerns by defining a natural (albeit highly stylized) generative model for

⁶E.g., the Open Directory Project <http://dmoz.org/>

the user distribution, which we then use in the experiments in Section 6. Given a tree metric space (X, \mathcal{D}) and the desired pointwise mean μ , the generative model provides a rich family of user distributions that are conditionally L-continuous w.r.t. $(X, c\mathcal{D})$, for some small c . We develop the generative model in Section 3.1. We extend this result in two directions: to arbitrary metric spaces (Section 3.2) and to distributions over conditionally L-continuous user distributions (Section 3.3).

3.1. Bayesian tree network

The generative model is a tree-shaped Bayesian network with 0-1 “relevance values” $\pi(\cdot)$ on nodes, where leaves correspond to documents. The tree is essentially a topical taxonomy on documents: subtopics correspond to subtrees. The relevance value on each subtopic is obtained from that on the parent topic via a low-probability mutation.

The mutation probabilities need to be chosen so as to guarantee conditional L-continuity and the desired pointwise mean μ . It is fairly easy to derive a necessary and sufficient condition for the pointwise mean, and a necessary condition for conditional L-continuity. The latter condition states that the mutation probabilities need to be bounded in terms of the distance between the child and the parent. The hard part is to prove that this condition is *sufficient*.

Let us describe our Bayesian tree network in detail. The network inputs a tree metric space (X, \mathcal{D}) and the desired pointwise mean μ , and outputs a relevance vector π . Specifically, we assume that documents are leaves of a finite rooted edge-weighted tree τ_d with node set V and leaf set $X \subset V$. Let \mathcal{D} be the (weighted) shortest-paths metric on V . The desired pointwise mean μ can be any function $\mu : X \rightarrow [\alpha, \frac{1}{2}]$, $\alpha > 0$, that is L-continuous w.r.t. (X, \mathcal{D}) . We show (see Section 7.1) that μ can be extended from X to V preserving the range and L-continuity.

The Bayesian network itself is very intuitive. We pick $\pi(\text{root}) \in \{0, 1\}$ at random with a suitable expectation, and then proceed top-down so that the child’s click is obtained from the parent’s click via a low-probability mutation. The mutation is parameterized by functions $q_0, q_1 : V \rightarrow [0, 1]$, as described in Algorithm 1. These parameters let us vary the degree of independence between each child and its parent, resulting in a rich family of user distributions.

To complete the construction, it remains to define the mutation probabilities q_0, q_1 . Let \mathcal{P} be the resulting user distribution. It is easy to see that μ is the point-

Algorithm 1 User distribution for tree metrics

Input: Tree (root r , node set V); $\mu(r) \in [0, 1]$
mutation probabilities $q_0, q_1 : V \rightarrow [0, 1]$

Output: relevance vector $\pi : V \rightarrow \{0, 1\}$

function AssignClicks(tree node v)

$b \leftarrow \pi(v)$
for each child u of v **do**
 $\pi(u) \leftarrow \begin{cases} 1 - b & \text{w/prob } q_b(u) \\ b & \text{otherwise} \end{cases}$
 AssignClicks(u)

Pick $\pi(r) \in \{0, 1\}$ at random with expectation $\mu(r)$
AssignClicks(r)

wise mean of \mathcal{P} on V if and only if

$$\mu(u) = (1 - \mu(v))q_0(u) + \mu(v)(1 - q_1(u)) \quad (7)$$

whenever u is a child of v . (For sufficiency, use induction on the tree.) Further, letting $q_b = q_b(u)$ for each bit $b \in \{0, 1\}$, note that

$$\begin{aligned} \Pr[\pi(u) \neq \pi(v)] &= \mu(v)q_1 + (1 - \mu(v))q_0 \\ &= \mu(v)(q_0 + q_1 + (1 - 2\mu(v))q_0) \\ &\geq \mu(v)(q_0 + q_1). \end{aligned}$$

Thus, if \mathcal{P} is L-correlated w.r.t. (X, \mathcal{D}) then

$$q_0(u) + q_1(u) \leq \mathcal{D}(u, v)/\mu(v). \quad (8)$$

We show that (7-8) suffices to guarantee conditional L-continuity. For a concrete example, one could define

$$(q_0(u), q_1(u)) = \begin{cases} \left(0, \frac{\mu(v) - \mu(u)}{\mu(v)}\right) & \text{if } \mu(v) \geq \mu(u) \\ \left(\frac{\mu(u) - \mu(v)}{1 - \mu(v)}, 0\right) & \text{otherwise.} \end{cases} \quad (9)$$

The provable properties of Algorithm 1 are summarized in the theorem below. It is technically more convenient to state this theorem in terms of L-correlation rather than L-continuity.

Theorem 3.1. *Let \mathcal{D} be the shortest-paths metric of an edge-weighted rooted tree with a finite leaf set X . Let $\mu : X \rightarrow [\alpha, \frac{1}{2}]$, $\alpha > 0$ be L-continuous w.r.t. (X, \mathcal{D}) . Suppose $q_0, q_1 : V \rightarrow [0, 1]$ satisfy (7-8).*

Let \mathcal{P} be the user distribution constructed by Algorithm 1. Then \mathcal{P} has pointwise mean μ and is conditionally L-correlated w.r.t. $(X, 3\mathcal{D}_\mu)$ where

$$\mathcal{D}_\mu(x, y) \triangleq \mathcal{D}(x, y) \min\left(\frac{1}{\alpha}, \frac{3}{\mu(x) + \mu(y)}\right). \quad (10)$$

Remark. The theorem can be strengthened by replacing \mathcal{D}_μ with the shortest-paths metric induced by \mathcal{D}_μ .

The proof of Theorem 3.1 is a key theoretical contribution of this work, and by far the most technical one. Below we provide a proof sketch. To keep the flow of the paper, the full proof is presented in Section 7.2.

Proof Sketch. As we noted above, the statement about the pointwise mean trivially follows from (7) using induction on the tree. In what follows we focus on conditional L-correlation.

Fix leaves $x, y \in X$ and a subset $S \subset X$. Let z be the least common ancestor of x and y . Recall that in Algorithm 1 the value of $\pi(\cdot)$ at each node is a random permutation of that of its parent. We focus on the event \mathcal{E} that no mutation happened on the $z \rightarrow x$ and $z \rightarrow y$ paths. Note that \mathcal{E} implies $\pi(x) = \pi(y) = \pi(z)$. Therefore

$$\Pr[\pi(x) \neq \pi(y) | Z_S] \leq \Pr[\bar{\mathcal{E}} | Z_S], \quad (11)$$

where $\bar{\mathcal{E}}$ is the negation of \mathcal{E} . Intuitively, $\bar{\mathcal{E}}$ is a low-probability “failure event”. The rest of the proof is concerned with showing that $\Pr[\bar{\mathcal{E}} | Z_S] \leq 3 \mathcal{D}_\mu(x, y)$.

First we handle the unconditional case. We claim that

$$\Pr[\bar{\mathcal{E}}] \leq \mathcal{D}_\mu(x, y). \quad (12)$$

Note that (12) immediately implies that \mathcal{P} is L-correlated w.r.t. (X, \mathcal{D}_μ) . This claim is not very difficult to prove, essentially since the condition (8) is specifically engineered to satisfy the unconditional L-correlation property. We provide the proof in detail.

Let $w \in \operatorname{argmin}_{u \in P_{xy}} \mu(u)$, where P_{xy} is the $x \rightarrow y$ path. Let $(z = x_0, x_1, \dots, x_n = x)$ be the $z \rightarrow x$ path. For each $i \geq 1$ by (8) the probability of having a mutation at x_i is at most $\mathcal{D}(x_i, x_{i-1})/\mu(w)$, so the probability of having a mutation on the $z \rightarrow x$ path is at most $\mathcal{D}(x, z)/\mu(w)$. Likewise for the $z \rightarrow y$ path. So $\Pr[\bar{\mathcal{E}}] \leq \mathcal{D}(x, y)/\mu(w) \leq \frac{1}{\alpha}$.

It remains to prove that

$$\Pr[\bar{\mathcal{E}}] \leq \mathcal{D}(x, y) \frac{3}{\mu(x) + \mu(y)}. \quad (13)$$

Indeed, by L-continuity it holds that

$$\begin{aligned} \mu(w) &\geq \mu(x) - \mathcal{D}(x, w), \\ \mu(w) &\geq \mu(y) - \mathcal{D}(y, w). \end{aligned}$$

Since $\mathcal{D}(x, y) = \mathcal{D}(x, w) + \mathcal{D}(y, w)$, it follows that

$$\mu(w) \geq \frac{\mu(x) + \mu(y) - \mathcal{D}(x, y)}{2}. \quad (14)$$

Now, either the right-hand side of (14) is at least $\frac{\mu(x) + \mu(y)}{3}$, or the right-hand side of (13) is at least 1. In both cases (13) holds. This completes the proof of the claim (12).

The conditional case is much more difficult. We handle it by showing that

$$\Pr[\bar{\mathcal{E}} | Z_S] \leq 3 \Pr[\bar{\mathcal{E}}]. \quad (15)$$

In fact, (15) holds even if (8) is replaced with a much weaker bound: $\max(q_0(u), q_1(u)) \leq \frac{1}{2}$ for each u .

The mathematically subtle proof of (15) can be found in Section 7.2. The crux in this proof is that event Z_S is more likely if document z is not relevant to the user:

$$\Pr[Z_S | z = 0] \geq \Pr[Z_S | z = 1]. \quad \square$$

3.2. Arbitrary metric spaces

We can extend Theorem 3.1 to arbitrary metric spaces using prior work on *metric embeddings*. Fix an N -point metric space (X, \mathcal{D}) and a function $\mu : X \rightarrow [\alpha, \frac{1}{2}]$ that is L-continuous on (X, \mathcal{D}) . It is known (Bartal, 1996; Fakcharoenphol et al., 2004) that there exists a distribution $\mathcal{P}_{\text{tree}}$ over tree metric spaces (X, \mathcal{T}) such that $\mathcal{D}(x, y) \leq \mathcal{T}(x, y)$ and

$$\mathbb{E}_{\mathcal{T} \sim \mathcal{P}_{\text{tree}}} [\mathcal{T}(x, y)] \leq c \mathcal{D}(x, y) \quad \forall x, y \in X,$$

where $c = O(\log N)$.⁷

The construction is simple: first sample a tree metric space (X, \mathcal{T}) from $\mathcal{P}_{\text{tree}}$, then independently generate a user distribution $\mathcal{P}_{\mathcal{T}}$ for (X, \mathcal{T}) as per Algorithm 1. Let \mathcal{P} be the resulting aggregate user distribution.

Theorem 3.2. *The aggregate user distribution \mathcal{P} has pointwise mean μ and is conditionally L-correlated w.r.t. $(X, 3c \mathcal{D}_\mu)$, where \mathcal{D}_μ is given by*

$$\mathcal{D}_\mu(x, y) = \mathcal{D}(x, y) \min \left(\frac{1}{\alpha}, \frac{3}{\mu(x) + \mu(y)} \right).$$

Proof. The function μ is L-continuous w.r.t. each tree metric space (X, \mathcal{T}) , so by Theorem 3.1 user distribution $\mathcal{P}_{\mathcal{T}}$ has pointwise mean μ and is conditionally L-correlated w.r.t. $(X, 3 \mathcal{T}_\mu)$. It follows that the aggregate user distribution \mathcal{P} has pointwise mean μ , and moreover for any $x, y \in X$ and $S \subset X$ we have

$$\begin{aligned} &\Pr_{\pi \sim \mathcal{P}} [\pi(x) \neq \pi(y) | Z_S] \\ &\leq \mathbb{E}_{\mathcal{T} \sim \mathcal{P}_{\text{tree}}} \left[\Pr_{\pi \sim \mathcal{P}_{\mathcal{T}}} [\pi(x) \neq \pi(y) | Z_S] \right] \\ &\leq \mathbb{E}_{\mathcal{T} \sim \mathcal{P}_{\text{tree}}} [3 \mathcal{T}_\mu(x, y)] \\ &\leq 3c \mathcal{D}_\mu(x, y). \end{aligned} \quad \square$$

⁷For point sets in a d -dimensional Euclidean space one could take $c = O(d \log \frac{1}{\epsilon})$, where ϵ is the minimal distance. In fact, this result extends to a much more general family of metric spaces – those of doubling dimension d . Doubling dimension, the smallest d such that any ball can be covered by 2^d balls of half the radius, is a well-studied concept in the theoretical computer science literature, e.g. see (Gupta et al., 2003).

3.3. Distributions over user distributions

Let us verify that a distribution over conditionally L-continuous user distributions is conditionally L-continuous. Apart from being a natural and desirable property, this result considerably extends the family of user distributions for which we have conditional L-continuity guarantees.

Lemma 3.3. *Let \mathcal{P} be a distribution over countably many user distributions \mathcal{P}_i that are conditionally L-continuous w.r.t. a metric space (X, \mathcal{D}) . Then \mathcal{P} is conditionally L-continuous w.r.t. (X, \mathcal{D}) .*

Proof. Let μ and μ_i be the (conditional) pointwise means of \mathcal{P} and \mathcal{P}_i , respectively. Formally, let us treat each \mathcal{P}_i as a measure, so that $\mathcal{P}_i(E)$ is the probability of event E under \mathcal{P}_i . Let $\mathcal{P} = \sum_i q_i \mathcal{P}_i$, where $\{q_i\}$ are positive coefficients that sum up to 1. Fix documents $x, y \in X$ and a subset $S \subset X$. Then

$$\begin{aligned} \mu(x|S) &= \mathcal{P}(x = 1 | Z_S) = \frac{\mathcal{P}(x = 1 \wedge Z_S)}{\mathcal{P}(Z_S)} \\ &= \frac{\sum_i q_i \mathcal{P}_i(x = 1 \wedge Z_S)}{\mathcal{P}(Z_S)} \\ &= \frac{\sum_i q_i \mathcal{P}_i(Z_S) \mu_i(x|Z_S)}{\mathcal{P}(Z_S)}. \end{aligned}$$

It follows that

$$\begin{aligned} |\mu(x|S) - \mu(y|S)| &= \frac{\sum_i q_i \mathcal{P}_i(Z_S) (\mu_i(x|Z_S) - \mu_i(y|Z_S))}{\mathcal{P}(Z_S)} \\ &\leq \frac{\sum_i q_i \mathcal{P}_i(Z_S) D(x, y)}{\mathcal{P}(Z_S)} \\ &\leq D(x, y). \end{aligned} \quad \square$$

4. Algorithms

In this section we present algorithms for the k -slot Lipschitz MAB problem defined in Section 2. In Sections 4.1-4.2 we describe several algorithms from prior work on, respectively, multi-slot MAB and Lipschitz MAB, and their adaptations to our setting. Interestingly, one can combine algorithms from prior work in a *modular* way, which we make particularly transparent by putting forward a suitable naming scheme. Then in Section 4.3 we review the *contextual bandit* setting and one particular algorithm for this setting. Finally, in Section 4.4 we represent our setting as a (multi-slot) contextual MAB problem, and use this representation to design two new algorithms that select documents slot-by-slot (top-down), and explicitly treat the higher selected documents as a *context* for the current slot.

As mentioned earlier, the “metric-aware” algorithms presented here are well-defined for arbitrary metric

spaces, but for simplicity we present them for a special case in which documents are leaves in a *document tree* τ_d with an ϵ -exponential tree metric. In all these algorithms, a *subtree* is chosen in each round. Then a document in this subtree is sampled at random, choosing uniformly at each branch.

4.1. Prior work: ranked bandits

Letting **Bandit** be some algorithm for the MAB problem, the “ranked” bandit algorithm **RankBandit** for the multi-slot MAB problem is defined as follows (Radlinski et al., 2008). We have k slots (i.e., ranks) for which we wish to find the best documents to present. In each slot i , a separate instance \mathcal{A}_i of **Bandit** is created. In each round these instances select the documents to show independently of one another. If a user clicks on slot i , then this slot receives a payoff of 1, and all higher (i.e., skipped) slots $j < i$ receive a payoff of 0. For slots $j > i$, the state is rolled back as if this round had never happened (as if the user never considered these documents). If no slot is clicked, then all slots receive a payoff of 0.

In (Radlinski et al., 2008), this approach gives rise to algorithms **RankUCB1** and **RankEXP3**, based on MAB algorithms **UCB1** and **EXP3** (Auer et al., 2002a, 2002b). **EXP3** is designed for the *adversarial* setting with no assumptions on how the clicks are generated, which translates into concrete provable guarantees for **RankEXP3**. **UCB1** is geared towards the *stochastic* setting with i.i.d. payoffs on each arm, although the per-slot i.i.d. assumption breaks for slots $i > 1$ because of the influence of the higher slots. Nevertheless, in small-scale experiments **RankUCB1** performs much better than **RankEXP3** (Radlinski et al., 2008).

In **UCB1**-style algorithms, including the zooming algorithm, one can damp exploration by replacing the $4 \log(T)$ factor in (16) with 1. Such change effectively makes the algorithm more *optimistic*; it was found beneficial for **RankUCB1** by Radlinski et al. (2008). We will denote this version by appending ‘+’ to the algorithm’s name, e.g. **RankUCB1+**. We will see that it can also greatly improve average performance here.

4.2. Prior work: using the metric space

Both above algorithms are impractical when there are too many documents to explore them all. To avoid this challenge, we can exploit the similarity information provided by the metric space in our setting. Since the payoff function $\mu(\cdot)$ is an L-continuous function on (X, \mathcal{D}) , we can use the (single-slot) bandit algorithms that are designed for the Lipschitz MAB problem to improve the speed of convergence of the **RankedBandit** approach.

Algorithm 2 “Zooming algorithm” in trees

initialize (document tree τ_d):
 $\mathcal{A} \leftarrow \emptyset$; activate($\text{root}(\tau_d)$)
activate($u \in \text{nodes}(\tau_d)$):
 $\mathcal{A} \leftarrow \mathcal{A} \cup \{u\}$; $n(u) \leftarrow 0$; $r(u) \leftarrow 0$
Main loop:
 $u \leftarrow \arg\max_{u \in \mathcal{A}} \frac{r(u)}{n(u)} + 2\text{rad}(u)$
 “Play” a random document from $\text{subtree}(u)$
 $r(u) \leftarrow r(u) + \{\text{reward}\}$; $n(u) \leftarrow n(u) + 1$
if $\text{rad}(u) < W(u)$ **then**
 deactivate u : remove u from \mathcal{A}
 activate all children of u

The meta-algorithm **GridBandit** (Kleinberg, 2004) is one such algorithm. It proceeds in phases: In phase i , the depth- i subtrees are treated as “arms”, and a fresh copy of **Bandit** is run on these arms.⁸ Phase i lasts for $k\epsilon^{-2i}$ rounds, where k is the number of depth- i subtrees. This meta-algorithm (coupled with an adversarial MAB algorithm such as **EXP3**) is the only prior algorithm that takes advantage of the metric space in the adversarial setting. Following (Radlinski et al., 2008), we expect **GridEXP3** to be overly pessimistic for our problem, trumped by the corresponding stochastic MAB approaches such as **GridUCB1**.

The “zooming algorithm” (Kleinberg et al., 2008b, Algorithm 2) is a more efficient version of **GridUCB1**: instead of iteratively reducing the grid size in the entire metric space, it selectively refines the grid in promising areas. It maintains a set \mathcal{A} of *active subtrees* which collectively partition the leaf set. In each round the active subtree with the maximal *index* is chosen. The index of a subtree is (assuming stochastic payoffs) the best available upper confidence bound on the click probabilities in this subtree. It is defined via the *confidence radius*⁹ given (letting T be the time horizon) by

$$\text{rad}(\cdot) \triangleq \sqrt{4 \log(T) / (1 + \#\text{samples}(\cdot))}. \quad (16)$$

The algorithm “zooms in” on a given active subtree u (de-activates u and activates all its children) when $\text{rad}(u)$ becomes smaller than its *width* $W(u) \triangleq \epsilon^{\text{depth}(u)} = \max_{x, x' \in u} \mathcal{D}(x, x')$. The “ranked zooming algorithm” will be denoted **RankZoom**.

4.3. Prior work: contextual bandits

Our subsequent algorithms leverage prior work on *contextual MAB*. The relevant contextual MAB setting is

⁸As an empirical optimization, previous events can also be replayed to better initialize later phases.

⁹The meaning of $\text{rad}(\cdot)$ is that the sample average is within $\pm \text{rad}(\cdot)$ from the true mean with high probability.

Algorithm 3 ContextZoom in trees

initialize (document tree τ_d , context tree τ_c):
 $\mathcal{A} \leftarrow \emptyset$; activate($\text{root}(\tau_d), \text{root}(\tau_c)$)
activate ($u \in \text{nodes}(\tau_d), u_c \in \text{nodes}(\tau_c)$):
 $\mathcal{A} \leftarrow \mathcal{A} \cup \{(u, u_c)\}$; $n(u, u_c) \leftarrow 0$; $r(u, u_c) \leftarrow 0$
Main loop:
 Input a context $h \in \text{nodes}(\tau_c)$
 $(u, u_c) \leftarrow \arg\max_{(u, u_c) \in \mathcal{A}: h \in u_c} W(u \times u_c) + \frac{r(u, u_c)}{n(u, u_c)} + \text{rad}(u, u_c)$
 “Play” a random document from $\text{subtree}(u)$
 $r(u, u_c) \leftarrow r(u, u_c) + \{\text{reward}\}$; $n(u, u_c) \leftarrow n(u, u_c) + 1$
if $\text{rad}(u, u_c) < W(u, u_c)$ **then**
 deactivate (u, u_c) : remove (u, u_c) from \mathcal{A}
 activate all pairs ($\text{child}(u), \text{child}(u_c)$)

as follows: in each round nature reveals a *context* h , an algorithm chooses a document x , and the resulting payoff is an independent $\{0, 1\}$ sample with expectation $\mu(x|h)$. Further, one is given *similarity information*: metrics \mathcal{D} and \mathcal{D}_c on documents and contexts, resp., such that for any two documents x, x' and any two contexts h, h' we have

$$|\mu(x|h) - \mu(x'|h')| \leq \mathcal{D}(x, x') + \mathcal{D}_c(h, h'). \quad (17)$$

Let X_c be the set of contexts, and $X_{dc} = X \times X_c$ be the set of all (document, context) pairs. Abstractly, one considers the metric space $(X_{dc}, \mathcal{D}_{dc})$, henceforth the *DC-space*, where the metric is

$$\mathcal{D}_{dc}((x, h), (x', h')) = \mathcal{D}(x, x') + \mathcal{D}_c(h, h').$$

We will use the “contextual zooming algorithm” (**ContextZoom**) from (Slivkins, 2009). This algorithm is well-defined for arbitrary \mathcal{D}_{dc} , but for simplicity we will state it for the case when \mathcal{D} and \mathcal{D}_c are ϵ -exponential tree metrics.

Let us assume that documents and contexts are leaves in a document tree τ_d and context tree τ_c , respectively. The algorithm (see Algorithm 3 for pseudocode) maintains a set \mathcal{A} of *active strategies* of the form (u, u_c) , where u is a subtree in τ_d and u_c is a subtree in τ_c . At any given time the active strategies partition X_{dc} . In each round, a context h arrives, and one of the active strategies (u, u_c) with $h \in u_c$ is chosen: namely the one with the maximal *index*, and then a document $x \in u$ is picked uniformly at random. The index of (u, u_c) is, essentially, the best available upper confidence bound on expected payoffs from choosing a document $x \in u$ given a context $h \in u_c$: as per (Slivkins, 2009), with high probability it holds that

$$\text{index}(u, u_c) \geq \mu(x|h), \quad \forall x \in u, h \in u_c. \quad (18)$$

The index is defined via sample average, confidence radius (16), and “width” $W(u \times u_c)$. The latter is an upper bound on the diameter of the “rectangle” $u \times u_c$ in the DC-space:

$$W(u, u_c) \geq \max_{x, x' \in u, h, h' \in u_c} \mathcal{D}(x, x') + \mathcal{D}_c(h, h'). \quad (19)$$

The (de)activation rule ensures that the active strategies form a finer partition in the regions of the DC-space that correspond to higher payoffs and more frequently occurring contexts.

4.4. New approach: ranked contextual bandits

We now present a new approach in which the upper slot selections are taken into account as a *context*.

The slot algorithms in the **RankBandit** setting can make their selections sequentially. Then without loss of generality each slot algorithm \mathcal{A}_i knows the set S of documents in the upper slots. We propose to treat S as a “context” to \mathcal{A}_i . Specifically, \mathcal{A}_i will assume that none of the documents in S is clicked, i.e. event Z_S happens (else the i -th slot is ignored by the user). For each such round, the click probabilities for \mathcal{A}_i are given by $\mu(\cdot | Z_S)$, which is an L-continuous function on (X, \mathcal{D}) .

Let us specify a suitable metric \mathcal{D}_c on contexts $S \subset X$:

$$\mathcal{D}_c(S, S') \triangleq 4 \inf \sum_{j=1}^n \mathcal{D}(x_j, x'_j), \quad (20)$$

where the infimum is taken over all $n \in \mathbb{N}$ and over all n -element sequences $\{x_j\}$ and $\{x'_j\}$ that enumerate, possibly with repetitions, all documents in S and S' .

We show that \mathcal{D}_c satisfies (17) via the following lemma.

Lemma 4.1. *Let $(X, \mathcal{D}, \mathcal{P})$ be an instance of the k -slot Lipschitz MAB problem. Fix $x \in X$ and $S, S' \subset X$. Define $\mathcal{D}_c(S, S')$ by (20). Then*

$$|\mu(x|Z_S) - \mu(x|Z_{S'})| \leq \mathcal{D}_c(S, S').$$

Proof. For shorthand, let us write

$$\begin{aligned} \sigma(x|S) &\triangleq 1 - \mu(x|Z_S), \\ \sigma(x|S, y) &\triangleq \sigma(x|S \cup \{y\}). \end{aligned}$$

First, we claim that for any $y \in X$ and $y' \in S$

$$|\sigma(x|S, y) - \sigma(x|S, y')| \leq 4 \mathcal{D}(y, y'). \quad (21)$$

Indeed, noting that $\sigma(x|S, y) = \sigma(y|S, x) \frac{\sigma(x|S)}{\sigma(y|S)}$, we

can re-write the left-hand side of (21) as

$$\begin{aligned} \text{LHS}(21) &= \sigma(x, S) \left| \frac{\sigma(y|S, x)}{\sigma(y|S)} - \frac{\sigma(y'|S, x)}{\sigma(y'|S)} \right| \\ &\leq \sigma(x, S) \mathcal{D}(y, y') \frac{\sigma(y|S) + \sigma(y'|S, x)}{\sigma(y|S) \sigma(y'|S)} \quad (22) \\ &= \mathcal{D}(y, y') \frac{\sigma(x|S) + \sigma(x|S, y)}{\sigma(y'|S)} \leq 2 \mathcal{D}(y, y'). \end{aligned}$$

In (22), we have used the L-continuity of $\sigma(\cdot|S)$ and $\sigma(\cdot|S, x)$. To achieve the constant of 2, it was crucial that $y' \in S$, so that $\sigma(y'|S) = 1$. This completes the proof of (21).

Fix some $n \in \mathbb{N}$ and some n -element sequences $\{x_i\}$ and $\{x'_i\}$ that enumerate, possibly with repetitions, all values in S and S' , respectively. Consider sets $S_i = \{x'_1, \dots, x'_i\} \cup \{x_{i+1}, \dots, x_n\}$, $1 \leq i \leq n-1$, and let $S_0 = S$ and $S_{n+1} = S'$. To prove the lemma, it suffices to show that

$$|\sigma(x|S_i) - \sigma(x|S_{i+1})| \leq 4 \mathcal{D}(x_{i+1}, x'_{i+1}) \quad (23)$$

for each $i \leq n$. To prove (23), fix i and let $y = x_{i+1}$ and $y' = x'_{i+1}$. Note that $S_i \cup \{y'\} = S_{i+1} \cup \{y\}$, call this set S^* . Then using (21) (note, $y \in S_i$ and $y' \in S'_i$) we obtain

$$\begin{aligned} |\sigma(x|S_i) - \sigma(x|S^*)| &= |\sigma(x|S_i, y) - \sigma(x|S_i, y')| \\ &\leq 2 \mathcal{D}(y, y'), \\ |\sigma(x|S_{i+1}) - \sigma(x|S^*)| &= |\sigma(x|S_{i+1}, y') - \sigma(x|S_{i+1}, y)| \\ &\leq 2 \mathcal{D}(y, y'), \end{aligned}$$

which implies (23). \square

Having defined \mathcal{D}_c , we can use any contextual MAB algorithm for \mathcal{A}_i . We will use **ContextZoom**.

Let us supply the missing details as they apply to the $(i+1)$ -th slot, $i \geq 1$. The contexts are unordered i -tuples of documents. Given a document tree τ_d , let us define *context tree* τ_c as follows. Depth- l nodes of τ_c are unordered i -tuples of depth- l nodes from τ_d , and leaves are contexts. The root of τ_c is $(r \dots r)$, where $r = \text{root}(\tau_d)$. For each internal node $u_c = (u_1 \dots u_i)$ of τ_c , its children are all unordered tuples $(v_1 \dots v_i)$ such that each v_j is a child of u_j in τ_d . This completes the definition of τ_c . Letting u and u_c be level- l subtrees of τ_d and τ_c , resp., it follows from (20) that $\mathcal{D}_c(S, S') \leq 4i \epsilon^l$ for any contexts $S, S' \in u_c$. Thus setting $W(u \times u_c) \triangleq \epsilon^l(4i+1)$ satisfies (19).

We will use **ContextZoom** (with τ_c and $W(u, u_c)$ as above) for slots $i \geq 2$; for slot 1, contexts are empty, so **ContextZoom** reduces to Algorithm 2. The resulting “ranked” algorithm is called **RankContextZoom**.

Exploiting correlations. We can further exploit the conditional L-continuity. In the analysis of **ContextZoom**, the index can be decreased, improving performance, as long as (18) holds. Fix context $S \subset X$. Since $\mu(y|Z_S) = 0$ for any $y \in S$,

$$\begin{aligned} \mu(x|Z_S) &= |\mu(x|Z_S) - \mu(y|Z_S)| \leq \mathcal{D}(x, y), \quad \forall y \in S \\ \text{hence } \mu(x|Z_S) &\leq \mathcal{D}(x, S) \triangleq \min_{y \in S} \mathcal{D}(x, y). \end{aligned} \quad (24)$$

In other words, if document x is close to some document in S , the event Z_S limits the conditional probability $\mu(x|Z_S)$. Using (24) we might be able to decrease the index of a strategy without violating (18). Thus, we “upgrade” **RankContextZoom** with the following *correlation rule*: for each active strategy (u, u_c) , where u is a subtree of documents, and u_c is a set of contexts, cap the index of each (u, u_c) at $\mathcal{D}(u, S)$.

We also define **RankCorrZoom**, a version of **RankZoom** which uses a similar “correlation rule”: for each slot, cap the index of each active subtree u at $\mathcal{D}(u, S)$. The intuition is (again) that decreasing the index provably increases performance of the zooming algorithm as long as the index is a valid upper bound on expected payoffs. We view **RankCorrZoom** as a light-weight way version of **RankContextZoom**: a low-overhead (but perhaps limited) way to use contexts inside **RankZoom**.

5. Provable scalability guarantees

Here we summarize the relevant provable guarantees from prior work, and apply them to our multi-slot algorithms. The purpose is two-fold: to provide intuition behind these algorithms, and to prove their scalability. We also provide an improved convergence result.

5.1. Prior work

Provable guarantees for single-slot MAB algorithms are usually expressed via *regret* w.r.t. a benchmark: the best arm in hindsight. Regret $R(T)$ of an algorithm is the expected payoff of the benchmark in T rounds minus that of the algorithm. We will discuss regret bounds for (i) the “plain” (unstructured) n -armed bandits, (ii) bandits on metric spaces, and (iii) contextual bandits. We will consider “adversarial payoffs” (with “oblivious”, non-adaptive adversary) and “stochastic payoffs” (with i.i.d. payoffs for each arm).

n -armed bandits. EXP3 (Auer et al., 2002b) achieves regret $R(T) = \tilde{O}(\sqrt{nT})$ against an oblivious (non-adaptive) adversary. In the stochastic setting, UCB1 (Auer et al., 2002a) performs much better, with *logarithmic* regret for every fixed μ . Namely, each arm $x \in X$ contributes only $O(\log T)/\Delta(x)$ to regret, where $\Delta(x) \triangleq \max \mu(\cdot) - \mu(x)$. Noting that the total regret from playing arms with $\Delta(\cdot) \leq \delta$ can be a priori

upper-bounded by δT , we bound regret of UCB1 as:

$$R(T) = \min_{\delta > 0} \left(\delta T + \sum_{x \in X: \Delta(x) > \delta} \frac{O(\log T)}{\Delta(x)} \right). \quad (25)$$

Note that (25) depends on μ . In particular, if $\Delta(\cdot) \geq \delta$ then $R(T) = O(\frac{n}{\delta} \log T)$. However, for any given T there exists a “worst-case” pointwise mean μ_T such that $R(T) = \tilde{\Theta}(\sqrt{nT})$ in (25), matching EXP3. The above regret guarantees for EXP3 and UCB1 are optimal up to constant factors (Auer et al., 2002b; Kleinberg et al., 2008a).

Bandits on metric spaces. For the Lipschitz MAB problem (Kleinberg, 2004; Kleinberg et al., 2008b), regret guarantees are independent of the number of arms. Instead, they depend on the covering properties of the metric space (X, \mathcal{D}) . The crucial notion here is the *covering number* $N_r(X)$, defined as the minimal number of balls of radius r sufficient to cover X . For metrics with “uniform density” (informally: without “denser” and “sparser” regions) the covering numbers can be usefully summarized by a single number called the “covering dimension”. Given a constant α , the *covering dimension* $\text{CovDim}(X, \mathcal{D})$ is defined as

$$\inf\{d \geq 0 : N_r \leq \alpha r^{-d} \quad \forall r > 0\}, \quad (26)$$

with $N_r = N_r(X)$.¹⁰ In particular, for an arbitrary point set in \mathbb{R}^d under the standard (ℓ_2) distance, the covering dimension is d , for some $\alpha = O(1)$. For an ϵ -exponential tree metric with maximal branching factor b , the covering dimension is $d = \log_{1/\epsilon}(b)$, with $\alpha = 1$.

Against an oblivious adversary, **GridEXP3** has regret

$$R(T) = \tilde{O}(\alpha T^{(d+1)/(d+2)}), \quad (27)$$

where d is the covering dimension of (X, \mathcal{D}) .

For the stochastic setting, **GridUCB1** and the zooming algorithm have better μ -specific regret guarantees similar to (25) for UCB1. In fact, we will state the guarantees for all three algorithms in a common form. Consider payoff scales $\mathcal{S} = \{2^i : i \in \mathbb{N}\}$, and for each scale $r \in \mathcal{S}$ define $X_r = \{x \in X : r < \Delta(x) \leq 2r\}$. Then regret (25) of UCB1 can be restated as

$$R(T) = \min_{\delta > 0} \left(\delta T + \sum_{r \in \mathcal{S}: r \geq \delta} N_{(\delta, r)} \frac{O(\log T)}{r} \right), \quad (28)$$

where $N_{(\delta, r)} = |X_r|$. In what follows, algorithms will have μ -specific regret bounds of the form (28), which will imply simpler (but sometimes less efficient) bounds of the form (27) (“dimension-type” bounds).

Now, it follows from the analysis in (Kleinberg, 2004; Kleinberg et al., 2008b) that regret of **GridUCB1** is (28)

¹⁰We will keep the constant α implicit in the notation.

with $N_{(\delta,r)} = N_\delta(X_r)$. For the worst-case μ one could have $N_\delta(X_r) = N_\delta(X)$, in which case the μ -specific bound essentially reduces to (27).

For the zooming algorithm, the μ -specific bound can be improved to (28) with $N_{(\delta,r)} = N_r(X_r)$. This implies an improved dimension-type bound: (27) with a different, smaller d called the *zooming dimension*, which is defined as (26) with $N_r = N_r(X_r)$. Note that the zooming dimension depends on the triple (X, \mathcal{D}, μ) rather than on the metric space alone.

The zooming dimension can be as high as the covering dimension for the worst-case μ , but can be much smaller (e.g., $d = 0$) for “benign” problem instances, see (Kleinberg et al., 2008b) for further discussion. For a simple example, suppose an ϵ -exponential tree metric has a “high-payoff” branch and a “low-payoff” branch with respective branching factors $b \ll b'$. Then the zooming dimension is $\log_{1/\epsilon}(b)$, whereas the covering dimension is $\log_{1/\epsilon}(b')$.

Contextual bandits. For the contextual MAB problem, the guarantees are in terms of *contextual regret*, which is regret w.r.t. a much stronger benchmark: the best arm in hindsight for every given context.

Regret guarantees for **ContextZoom** focus on the *DC-space* $(X_{dc}, \mathcal{D}_{dc})$ (see Section 4.3 for notation). We partition X_{dc} according to payoff scales $r \in \mathcal{S}$:

$$\Delta(x|h) \triangleq \max \mu(\cdot|h) - \mu(x|h), \quad x \in X, h \in X_c.$$

$$X_{dc,r} \triangleq \{(x, h) \in X_{dc} : r < \Delta(x|h) \leq 2r\}.$$

Then contextual regret of **ContextZoom** can be bounded by (28) with $N_{(\delta,r)} = N_r(X_{dc,r})$, where $N_r(\cdot)$ now refers to the covering numbers in the DC-space. To derive the corresponding dimension-type bound, define the *contextual zooming dimension* d_{dc} of the problem instance $(X_{dc}, \mathcal{D}_{dc}, \mu)$ as (26) with $N_r = N_r(X_{dc,r})$. Then one obtains (27) with $d = d_{dc}$. It is easy to see that

$$\text{CovDim}(X_c, \mathcal{D}_c) \leq d_{dc} \leq \text{CovDim}(X_{dc}, \mathcal{D}_{dc}). \quad (29)$$

The upper bound in (29) corresponds to the worst-case μ such that $N_r(X_{dc,r}) = N_r(X_{dc})$.

The μ -specific bound for **ContextZoom** can be improved by taking into account “benign” context arrivals: effectively, one can prune the regions of X_c that correspond to infrequent context arrivals, see (Slivkins, 2009) for details. This improvement can be especially significant if $\text{CovDim}(X_c, \mathcal{D}_c) > \text{CovDim}(X, \mathcal{D})$.

Discussion 5.1. **ContextZoom** is parameterized by the time horizon, denote it T_0 . To obtain a regret bound that holds for each round $T \leq T_0$, replace the $\log(T)$ term in (28) with $\log(T_0)$ (which then implies the corresponding dimension-type bound). In fact, this regret

bound holds with probability at least $1 - T_0^{-2}$, so that one can take the Union Bound over all rounds $T \leq T_0$. Moreover, the *anytime* version of **ContextZoom** can be defined via the *doubling trick*: in each phase $i \in \mathbb{N}$, run a fresh instance of the algorithm for 2^i rounds. This version is well-defined (and, essentially, has the same regret bounds) for any time T . This discussion also applies to the zooming algorithm.

Multi-slot bandits. Letting T be the number of rounds and OPT be the probability of clicking on the optimal ranking, algorithm **RankBandit** achieves

$$\mathbb{E}[\#\text{clicks}] \geq (1 - \frac{1}{e})T \times \text{OPT} - k R(T), \quad (30)$$

where $R(T)$ is any upper bound on the context-oblivious regret for **Bandit** in each slot (Radlinski et al., 2008).

In the multi-slot setting, *performance* of an algorithm up to time t is defined as the time-averaged expected total number of clicks. We will consider performance as a function of t . Assuming $R(T) = o(T)$ in (30), performance of **RankBandit** converges to or exceeds $(1 - \frac{1}{e})\text{OPT}$. Convergence to $(1 - \frac{1}{e})\text{OPT}$ is proved to be worst-case optimal. Thus, as long as $R(T)$ scales well with $\#\text{documents}$ (e.g., as in (27)), Radlinski et al. (2008) interpret (30) as a proof of an algorithm’s scalability in the multi-slot MAB setting.

RankBandit is presented in (Radlinski et al., 2008) as the online version of the *greedy algorithm*: an offline fully informed algorithm that selects documents greedily slot by slot from top to bottom. The performance of this algorithm is called the *greedy optimum*,¹¹ which is equal to $(1 - \frac{1}{e})\text{OPT}$ in the worst case, but for “benign” problem instances it can be as good as OPT . The greedy optimum is a more natural benchmark for **RankBandit** than $(1 - \frac{1}{e})\text{OPT}$. Surprisingly, results w.r.t. this benchmark are absent in the literature.

5.2. Our results

Regret. We show that **RankGridEXP3** and **RankContextZoom** scale to large document collections, in the sense that they achieve (30) with $R(T)$ that does not degenerate with $\#\text{documents}$.

Theorem 5.2. *Let $(X, \mathcal{D}, \mathcal{P})$ be an instance of the k -slot Lipschitz MAB problem. Then*

- (a) **RankGridEXP3** achieves regret (30), where $R(T)$ is given by (27) with $d = \text{CovDim}(X, \mathcal{D})$.
- (b) Let d_{dc} be the contextual zooming dimension for slot i . Then **RankContextZoom** achieves (30), where $R(T)$ is given by (27) with $d = d_{dc}$.

¹¹If due to ties there are multiple “greedy rankings”, define the greedy optimum via the *worst* of them.

Proof. The respective regret bounds for **RankGridEXP3** and **RankContextZoom** plug into (30). \square

Remark 5.3. We do not have any provable guarantees for **RankGridUCB1**, **RankZoom** and **RankCorrZoom** because non-contextual regret bounds for the single-slot stochastic setting do not plug into (30).

Discussion 5.4. Contrary to our intuition, our provable guarantee for **RankContextZoom** is *worse* than the one for **RankGridEXP3**. Indeed, by (29) it holds that

$$d_{dc} \geq \text{CovDim}(X_c, \mathcal{D}_c) = k \times \text{CovDim}(X, \mathcal{D}),$$

where (X_c, \mathcal{D}_c) is the context space for slot k .

However, we believe that the bound in part (b) is very pessimistic. It builds on the result for **ContextZoom** that contextual regret for slot i is bounded by (27) with $d = d_{dc}$. This result does not take into account that for a given slot, contexts $S \subset X$ may gradually converge over time to the greedy optimum and thus become “low-dimensional”.¹² We believe this effect is very important to the performance **RankContextZoom**. In particular, it causes **RankContextZoom** to perform much better than **RankGridEXP3** in simulations.

Benchmark. Recall that while the bound in (30) uses $(1 - \frac{1}{e})\text{OPT}$ as a benchmark, a more natural benchmark would be the greedy optimum. We provide a preliminary convergence result for **RankContextZoom**, without any specific regret bounds. Such result is more elegantly formulated in terms of “anytime-**RankContextZoom**” which uses the “anytime” version of **ContextZoom** (see Discussion 5.1).

Theorem 5.5. *Fix an instance of the k -slot MAB problem. The performance of anytime-**RankContextZoom** up to any given time t is equal to the greedy optimum minus $f(t)$ such that $f(t) \rightarrow 0$.*

Proof Sketch. It suffices to prove that with high probability, anytime-**RankContextZoom** outputs a greedy ranking in all but $f_k(t)$ rounds among the first t rounds, where $f_k(t) \rightarrow 0$.

We prove this claim by induction on k , the number of slots. Suppose it holds for some $k - 1$ slots, and focus on the k -th slot. Consider all rounds in which a greedy ranking is chosen for the upper slots but not for the k -th slot. In each such round, the k -th slot replica of anytime-**ContextZoom** incurs contextual regret at least δ_k , for some instance-specific constant $\delta_k > 0$. Thus, with high probability there can be at most $R_k(t)/\delta_k$ such rounds, where $R_k(t) = o(t)$ is an upper bound on contextual regret for slot k . Thus, one can take $f_k(t) = f_{k-1}(t) + R_k(t)/\delta_k$. \square

¹²It is also wasteful (but perhaps less so) that we use a slot- k bound for each slot $i < k$.

Remark 5.6. Theorem 5.5 is about the “metric-less” setting from (Radlinski et al., 2008). It easily extends to the “ranked” version of any bandit algorithm whose contextual regret is sublinear with high probability.

Discussion 5.7. It is an open question whether (and under which assumptions) Theorem 5.5 can be extended to the “ranked” versions of non-contextual bandit algorithms such as **RankUCB1**. One assumption that appears essential is the uniqueness of the greedy ranking. To see that multiple greedy rankings may cause problems for ranked non-contextual algorithms, consider a simple example:

- There are two slots and three documents x_1, x_2, x_3 such that $\mu = (\frac{1}{2}, \frac{1}{2}, \frac{1}{3})$ and the relevance of each arm is independent of that of the other arms.¹³

An optimal ranking for this example is a greedy ranking that puts x_1 and x_2 in the two slots, achieving aggregate click probability $\frac{3}{4}$. According to our intuition, a “reasonable” ranked non-contextual algorithm will behave as follows. The slot 1 algorithm will alternate between x_1 and x_2 , each with frequency $\rightarrow \frac{1}{2}$. Since the slot-2 algorithm is oblivious to the slot 1 selection, it will observe averages that converge over time to $(\frac{1}{4}, \frac{1}{4}, \frac{1}{3})$,¹⁴ so it will select document x_3 with frequency $\rightarrow 1$. Therefore frequency $\rightarrow 1$ the ranked algorithm will alternate between (x, z) or (y, z) , each of which has aggregate click probability $\frac{2}{3}$.

Desiderata. We believe that the above guarantees do not reflect the full power of our algorithms, and more generally the full power of conditional L-continuity. The “ideal” performance guarantee for **RankBandit** in our setting would use the greedy optimum as a benchmark, and would have a bound on regret that is free from the inefficiencies outlined in Discussion 5.4. Furthermore, this guarantee would only rely on some general property of **Bandit** such as a bound on (contextual) regret. We conjecture that such guarantee is possible for **RankContextZoom**, and (under some assumptions) for **RankCorrZoom** and **RankZoom**.

Further, one would like to study the relative benefits of the new “contextual” algorithms (**RankContextZoom** and **RankCorrZoom**) and the prior work (e.g., **RankZoom**). Discussion 5.7 suggests that the difference can be particularly pronounced when the pointwise mean has multiple peaks of similar value. We confirm this

¹³To make it more challenging for an algorithm, each document x_j can stand for a disjoint *subset* X_j of documents with highly correlated payoffs. Further, some documents in X_j can be far apart in the metric space.

¹⁴Suppose $x_j, j \in \{1, 2\}$ is chosen in slot 1. Then, letting $S = \{x_j\}$, $\mu(x_1|Z_S)$ equals 0 if $j = 1$ and $\frac{1}{2}$ otherwise (which averages to $\frac{1}{4}$), whereas $\mu(x_3|Z_S) = \frac{1}{3}$.

experimentally in Section 6.3.

6. Evaluation

Let us evaluate the performance of the algorithms presented in Section 4: “metric-oblivious” **RankUCB1** and **RankEXP3**, “metric-aware” non-contextual **RankGridUCB1**, **RankGridEXP3** and **RankZoom**, and contextual **RankContextZoom** and **RankCorrZoom**.

6.1. Experimental setup

Using the generative model from Section 3 (Algorithm 1 with (9)), we created a document collection with $|X| = 2^{15} \approx 32,000$ documents¹⁵ in a binary ϵ -exponential tree metric space with $\epsilon = 0.837$. The value for ϵ was chosen so that the most dissimilar documents in the collection still have a non-trivial similarity, as may be expected for web documents. Each document’s expected relevance $\mu(x)$ was set by first identifying a small number of “peaks” $y_i \in X$, choosing $\mu(\cdot)$ for these documents, and then defining the relevance of other documents as the minimum allowed while obeying L-continuity and a background relevance rate μ_0 :

$$\mu(x) \triangleq \max(\mu_0, \frac{1}{2} - \min_i \mathcal{D}(x, y_i)). \quad (31)$$

For internal nodes in the tree, μ is defined bottom-up (from leaves to the root) as the mean value of all children nodes. As a result, we obtain a set of documents X where each document $x \in X$ has an expected click probability $\mu(x)$ that obeys L-continuity.

Our simulation was run over a 5-slot ranked bandit setting, learning the best 5 documents. We evaluated over 300,000 user visits sampled from \mathcal{P} per Algorithm 1. Performance within 50,000 impressions, typical for the number of times relatively frequent queries are seen by commercial search engines in a month, is essential for any practical applicability of this approach. However, we also measure performance for a longer time period to obtain a deeper understanding of the convergence properties of the algorithms.

We consider two models for $\mu(\cdot)$ in (31). In the first model, two “peaks” $\{y_1, y_2\}$ are selected at random with $\mu(\cdot) = \frac{1}{2}$, and μ_0 set to 0.05. The second model is less “rigid” (and thus more realistic): the relevant documents y_i and their expected relevance rates $\mu(\cdot)$ are selected according to a Chinese Restaurant Process (Aldous, 1985) with parameters $n=20$ and $\theta=2$, and setting $\mu_0 = 0.01$. The Chinese Restaurant Process is inspired by customers coming in to a restaurant with an infinite number of tables, each with infinite capac-

ity. At time t , a customer arrives and can choose to sit at a new table with probability $\theta/(t-1+\theta)$, and otherwise sits at an already occupied table with probability proportional to the number of customers already sitting at that table. By considering each table as equivalent to a peak in the distribution, this leads to a set of peaks with expected relevance rates distributed according to a power law. Following (Radlinski et al., 2008), we assign users to one of the peaks, then select relevant documents so as to obey the expected relevance rate $\mu(x)$ for each document x .

As baselines we use an algorithm ranking the documents at random, and the (offline) greedy algorithm discussed in Section 5.1.

6.2. Experimental results

Our experimental results are summarized in Figure 1 and Figure 2.

First, **RankEXP3** and **RankUCB1** perform as poorly as picking documents randomly: the three curves are indistinguishable. This is due to the large number of available documents and slow convergence rates of these algorithms. Other algorithms that explore all strategies (such as REC (Radlinski et al., 2008)) would perform just as poorly. This result is consistent with results reported by (Radlinski et al., 2008) on just 50 documents. On the other hand, algorithms that progressively refine the space of strategies explored perform much better.

Second, Making the UCB1-style algorithms “optimistic” (marked with a “+” after the algorithm name, see Section 4.1 for a discussion) improved performance dramatically. In particular, **GridUCB1+** performed best of the non-contextual algorithms. **RankZoom** performs comparably to optimistic **RankUCB1**, and becomes extremely effective if made optimistic. For the two contextual algorithms, we saw a similar increase in performance, hence we only show the performance of the optimistic versions.

Third, **RankCorrZoom+** achieves the best empirical performance, converging rapidly to near-optimal rankings. **RankZoom+** is a close second. Interestingly, the theoretically preferred **RankContextZoom** does not perform as well in simulations. This appears to be due to the much larger branching factor in the strategies activated by **RankContextZoom** slowing down the convergence.

6.3. Secondary experiment

As discussed in Section 5 (Discussion 5.7), some **RankBandit**-style algorithms may converge to a suboptimal ranking if μ has multiple peaks with similar values. To

¹⁵This is a realistic number of documents that may be considered in detail for a typical web search query after pruning very unlikely documents.

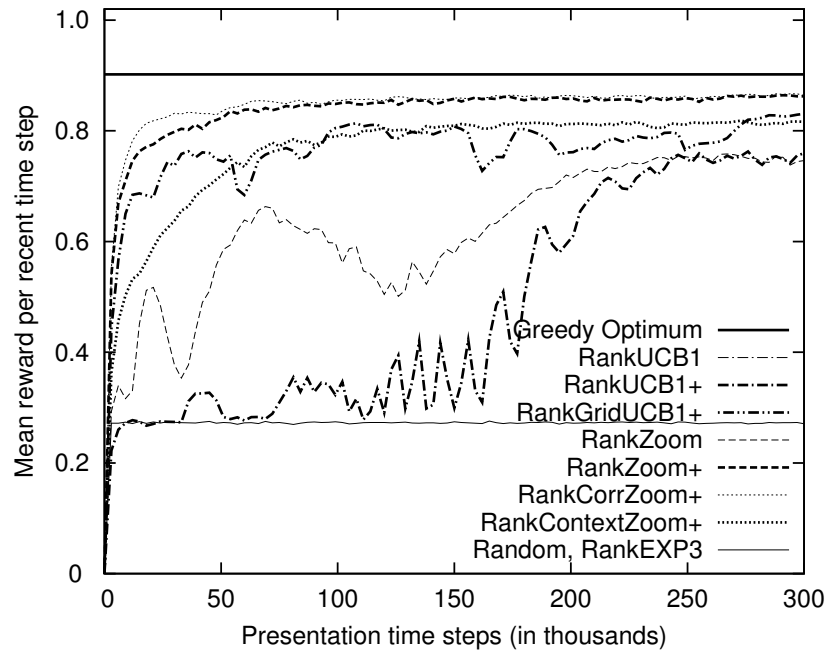


Figure 1. The various learning algorithms on 5-slot problem instances with two relevance peaks.

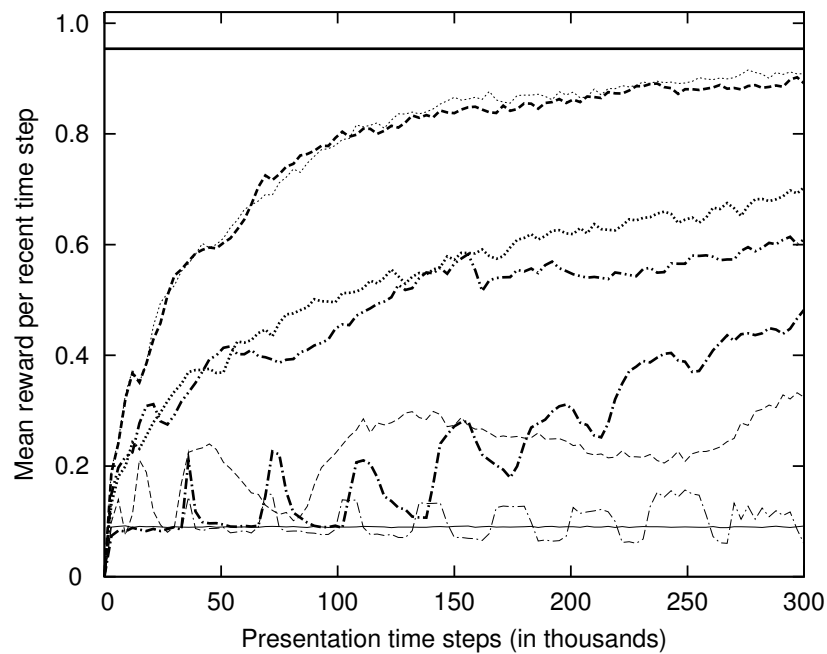


Figure 2. The various learning algorithms on 5-slot problem instances with random relevance rates $\mu(\cdot)$ selected according to the Chinese Restaurant Process.

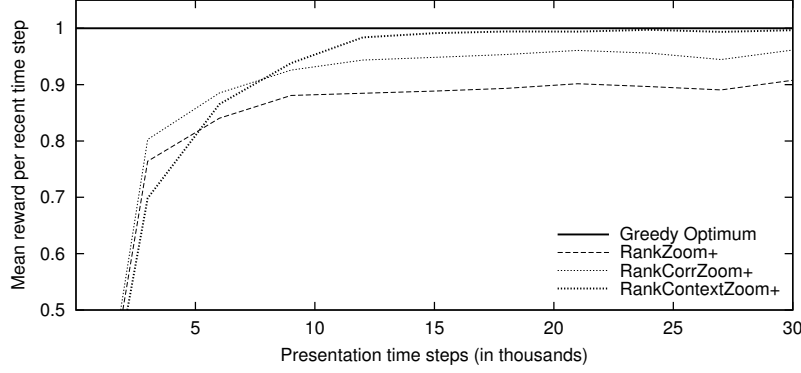


Figure 3. Comparison of zooming variants in a two-slot setting over a small document collection.

investigate this, we designed a small-scale experiment presented in Figure 3. We generated a small collection of 128 documents using the same setup with two “peaks”, and assumed 2 slots. Each peak corresponds to a half of the user population, with peak value $\mu = \frac{1}{2}$ and background value $\mu_0 = 0.05$.

We see that **RankContextZoom+** converges more slowly than the other zooming variants, but eventually outperforms them. This confirms our intuition, and suggests that **RankContextZoom+** may eventually outperform the other algorithms on a larger collection, such as that used for Figures 1 and 2.

7. Proof of Theorem 3.1

This section completes the proof of Theorem 3.1. First we state and prove a lemma needed to make the generative model well-defined. Then we complete the proof sketch in Section 3.1 by proving (15).

7.1. Extending μ from leaves to tree nodes

Implicit in the construction in Section 3 is the following lemma which extends μ from documents (leaves of the document tree) to the internal nodes of the tree.

Lemma 7.1. *Let \mathcal{D} be the shortest-paths metric of an edge-weighted rooted tree with node set V and leaf set X . Let $\mu : X \rightarrow [a, b]$ be an L -continuous function on (X, \mathcal{D}) . Then μ can be extended to V so that $\mu : V \rightarrow [a, b]$ is L -continuous w.r.t. (V, \mathcal{D}) .*

Proof. For each $x \in V$, let $\mathcal{L}(x)$ be the set of all leaves in the subtree rooted at x . For each $z \in \mathcal{L}(y)$ the assignment $\mu(x)$ should satisfy

$$\mu(z) - \mathcal{D}(x, z) \leq \mu(x) \leq \mu(z) + \mathcal{D}(x, z)$$

Thus $\mu(x)$ should lie in the interval $I(x) \triangleq$

$[\mu^-(x), \mu^+(x)]$, where

$$\mu^-(x) \triangleq \sup_{z \in \mathcal{L}(x)} \mu(z) - \mathcal{D}(x, z),$$

$$\mu^+(x) \triangleq \inf_{z \in \mathcal{L}(x)} \mu(z) + \mathcal{D}(x, z).$$

This interval is always well-defined, i.e. $\mu^-(x) \leq \mu^+(x)$. Indeed, if not then for some $z, z' \in \mathcal{L}(x)$

$$\mu(z) - \mathcal{D}(x, z) > \mu(z') + \mathcal{D}(x, z')$$

$$\mu(z) - \mu(z') > \mathcal{D}(x, z) + \mathcal{D}(x, z') \geq \mathcal{D}(z, z'),$$

contradiction, claim proved. Note that $\mu^+(x) \geq a$ and $\mu^-(x) \leq b$, so the intervals $I(x)$ and $[a, b]$ overlap.

Using induction on the tree, we will construct values $\mu(x)$, $x \in V$ such that the Lipschitz condition

$$|\mu(x) - \mu(y)| \leq \mathcal{D}(x, y) \quad \text{for all } x, y \in X$$

holds whenever x is a parent of y . For the root x_0 , let $\mu(x_0)$ be an arbitrary value in the interval $I(x_0) \cap [a, b]$. For the induction step, suppose for some x we have chosen $\mu(x) \in I(x) \cap [a, b]$ and y is a child of x . We need to choose $\mu(y) \in I(y) \cap [a, b]$ so that $|\mu(x) - \mu(y)| \leq \mathcal{D}(x, y)$. Note that

$$\begin{aligned} \mu(x) &\geq \mu^-(x) \geq \sup_{z \in \mathcal{L}(y)} [\mu(z) - \mathcal{D}(x, y) - \mathcal{D}(y, z)] \\ &= \mu^-(y) - \mathcal{D}(x, y), \end{aligned}$$

$$\begin{aligned} \mu(x) &\leq \mu^+(x) \leq \inf_{z \in \mathcal{L}(y)} [\mu(z) + \mathcal{D}(x, y) + \mathcal{D}(y, z)] \\ &= \mu^+(y) + \mathcal{D}(x, y). \end{aligned}$$

It follows that $I(y)$ and $[\mu(x) - \mathcal{D}(x, y), \mu(x) + \mathcal{D}(x, y)]$ have a non-empty intersection. Therefore, both intervals have a non-empty intersection with $[a, b]$. So we can choose $\mu(y)$ as required. This completes the construction of $\mu(\cdot)$ on V .

To check that μ is Lipschitz-continuous on V , fix $x, y \in V$, let P be the $x \rightarrow y$ path in the tree, and note that

$$\begin{aligned} |\mu(x) - \mu(y)| &\leq \sum_{(u,v) \in P} |\mu(u) - \mu(v)| \\ &\leq \sum_{(u,v) \in P} \mathcal{D}(u, v) = \mathcal{D}(x, y). \quad \square \end{aligned}$$

7.2. Conditional L-correlation: proof of (15)

Let us re-introduce the notation from the proof sketch, in slightly more detail. Fix documents $x, y \in X$. We focus on the key event, denoted \mathcal{E} , that no mutation happened on the $x \rightarrow y$ path. Recall that in Algorithm 1, for each tree node u with parent v we assign $\pi(u) \leftarrow M_u(\pi(v))$, where $M_u : \{0, 1\} \rightarrow \{0, 1\}$ is a random mutation which flips the input bit b with probability $q_b(u)$. If M_u is the identity function, then we say that no mutation happened at u . We say that no mutation happened on the $x \rightarrow y$ path if no mutation happened at each node in N_{xy} , the set of all nodes on the $x \rightarrow y$ path except z . This event is denoted \mathcal{E} ; note that it implies $\pi(x) = \pi(y) = \pi(z)$. Its complement $\bar{\mathcal{E}}$ is, intuitively, a low-probability “failure event”.

Fix a subset of documents $S \subset X$. Recall that Z_S denotes the event that all documents in S are irrelevant, i.e. $\pi(\cdot) = 0$ on S . Then (15) follows from the following lemma:

Lemma 7.2. $\Pr[\bar{\mathcal{E}} \mid Z_S] \leq \Pr[\bar{\mathcal{E}}] \times (2 / \Pr[\mathcal{E}])$.

Indeed, letting $p = \Pr[\bar{\mathcal{E}}]$ it holds that

$$\Pr[\bar{\mathcal{E}} \mid Z_S] \leq \min\left(1, \frac{2p}{1-p}\right) \leq 3p.$$

Remark. Lemma 7.2 inherits assumptions (7-8) on the mutation probabilities. For this Lemma, the upper bound (7) on mutation probabilities can be replaced with a much weaker one:

$$\max(q_0(u), q_1(u)) \leq \frac{1}{2} \quad \text{for each tree node } u. \quad (32)$$

In the rest of this section we prove Lemma 7.2. The proof is detailed, and quite heavy on notation. Curiously, we found this proof more intuitive if one actually *thinks* in terms of this (or similar) notation.

Let \mathcal{T}_u be the subtree rooted at a tree node u . For convenience, we will write $u = b$, $b \in \{0, 1\}$ to mean $\pi(u) = b$.

In a sequence on claims, we will establish that

$$\Pr[Z_S \mid z = 0] \geq \Pr[Z_S \mid z = 1]. \quad (33)$$

Intuitively, (33) means that the low-probability mutations are more likely to zero out a given subset of the leaves if the value at some fixed internal node is zero (rather than one).

Before we prove (33), we use it to derive Lemma 7.2.

7.2.1. USING (33) TO PROVE LEMMA 7.2

Let us extend the notion of mutation from a single node to the $x \rightarrow y$ path. Recall that N_{xy} denotes

the set of all nodes on this path except z . Then the individual node mutations $\{M_u : u \in N_{xy}\}$ collectively provide a mutation on N_{xy} , which we define simply as a function $M : N_{xy} \times \{0, 1\} \rightarrow \{0, 1\}$ such that $\pi(\cdot) = M(\cdot, \pi(z))$. Crucially, M is chosen independently of $\pi(z)$ (and of all other mutations). Let \mathcal{M} be the set of all possible mutations of N_{xy} . By a slight abuse of notation, we treat the event \mathcal{E} as the identity mutation.

Claim 7.3. Fix $M \in \mathcal{M}$ and $b \in \{0, 1\}$. Then

$$\Pr[Z_S \mid M, \pi(z) = b] \leq \Pr[Z_S \mid \mathcal{E}, \pi(z) = 0].$$

Proof. For each tree node u , let $S_u = S \cap \mathcal{T}_u$ be the subset of S that lies in the subtree \mathcal{T}_u . Then by (33)

$$\begin{aligned} \Pr[Z_S \mid M, \pi(z) = b] &= \prod_u \Pr[Z_{S_u} \mid \pi(u) = M(u, b)] \\ &\leq \prod_u \Pr[Z_{S_u} \mid \pi(u) = 0] \\ &= \Pr[Z_S \mid \mathcal{E}, \pi(z) = 0], \end{aligned}$$

where the product is over all tree nodes $u \in N_{xy}$ such that the intersection S_u is non-empty. \square

Proof of Lemma 7.2. On one hand, by Claim 7.3

$$\begin{aligned} \Pr[Z_S \cap \bar{\mathcal{E}}] &= \sum_{b, M} \Pr[M] \Pr[z = b] \Pr[Z_S \mid M, z = b] \\ &\leq \sum_{b, M} \Pr[M] \Pr[z = b] \Pr[Z_S \mid \mathcal{E}, z = 0] \\ &= \Pr[\bar{\mathcal{E}}] \times \Pr[Z_S \mid \mathcal{E}, z = 0], \end{aligned}$$

where the sums are over bits $b \in \{0, 1\}$ and all mutations $M \in \mathcal{M} \setminus \{\mathcal{E}\}$. On the other hand,

$$\Pr[Z_S] = \sum_{b, M} \Pr[M] \Pr[z = b] \Pr[Z_S \mid M, z = b]$$

(where the sum is over $b \in \{0, 1\}$ and $M \in \mathcal{M}$)

$$\geq \Pr[\mathcal{E}] \Pr[z = 0] \Pr[Z_S \mid \mathcal{E}, z = 0].$$

Since $\Pr[z = 0] \geq \frac{1}{2}$, it follows that

$$\begin{aligned} \Pr[\bar{\mathcal{E}} \mid Z_S] &= \Pr[Z_S \cap \bar{\mathcal{E}}] / \Pr[Z_S] \\ &\leq 2 \Pr[\bar{\mathcal{E}}] / \Pr[\mathcal{E}]. \end{aligned} \quad \square$$

7.2.2. PROOF OF (33)

First we prove (33) for the case $S \subset \mathcal{T}_z$, then we build on it to prove the (similar, but considerably more technical) case $S \cap \mathcal{T}_z = \emptyset$. The general case follows since the events $Z_{S \cap \mathcal{T}_z}$ and $Z_{S \setminus \mathcal{T}_z}$ are conditionally independent given $\pi(z)$.

Claim 7.4. If $S \subset \mathcal{T}_z$ then (33) holds.

Proof. Let us use induction the depth of z . For the base case, the case $x = y = z$. Then $S = \{z\}$ is the only possibility, and the claim is trivial.

For the induction step, consider children u_i of z such that the intersection $S_i \triangleq S \cap \mathcal{T}_{u_i}$ is non-empty. Let u_1, \dots, u_k be all such children. For brevity, denote $Z_i \triangleq Z_{S_i}$, and

$$\nu_i(a|b) \triangleq \Pr[u_i = a | z = b], \quad a, b \in \{0, 1\}.$$

Note that $\nu_i(1, 0) = q_0(x_i)$ and $\nu_i(0, 1) = q_1(x_i)$.

Then for each $b \in \{0, 1\}$ we have

$$\Pr[Z_S | z = b] = \prod_{i=1}^k \Pr[Z_i | z = b] \quad (34)$$

$$\Pr[Z_i | z = b] = \sum_{a \in \{0, 1\}} \nu_i(a|b) \Pr[Z_i | u_i = a]. \quad (35)$$

By (34), to prove the claim it suffices to show that

$$\Pr[Z_i | z = 0] \geq \Pr[Z_i | z = 1] \quad (36)$$

holds for each i . By the induction hypothesis we have

$$\Pr[Z_i | u_i = 0] \geq \Pr[Z_i | u_i = 1]. \quad (37)$$

Combining (37) and (32), and noting that by (35) we have $\nu_i(0|0) \geq \nu_i(0|1)$, it follows that

$$\begin{aligned} & \Pr[Z_i | z = 0] - \Pr[Z_i | z = 1] \\ &= \sum_{a \in \{0, 1\}} \Pr[Z_i | u_i = a] (\nu_i(a|0) - \nu_i(a|1)) \\ &\geq \Pr[Z_i | u_i = 1] \sum_{a \in \{0, 1\}} (\nu_i(a|0) - \nu_i(a|1)) \\ &= 0 \end{aligned}$$

because $\nu_i(0|0) + \nu_i(1|0) = \nu_i(0|1) + \nu_i(1|1) = 1$. \square

Corollary 7.5. *Consider tree nodes r, v, w such that r is an ancestor of v which in turn is an ancestor of w . Then for any $c \in \{0, 1\}$*

$$\Pr[u = 0 | w = 0, r = c] \geq \Pr[u = 0 | w = 1, r = c].$$

Proof. We claim that for each $b \in \{0, 1\}$

$$\Pr[w = b | u = b] \geq \Pr[w = b | u = 1 - b]. \quad (38)$$

Indeed, truncating the subtree \mathcal{T}_w to a single node w and specializing Lemma 7.4 to a singleton set $S = \{w\}$ (with $z = u$) we obtain (38) for $b = 0$. The case $b = 1$ is symmetric.

Now, for brevity we will omit conditioning on $\{r = c\}$ in the remainder of the proof. (Formally, we will work on in the probability space obtained by conditioning on this event.) Then for each $b \in \{0, 1\}$

$$\begin{aligned} & \Pr[u = 0 | w = b] \\ &= \frac{\Pr[u = 0 \wedge w = b]}{\Pr[u = 0 \wedge w = b] \cup \Pr[u = 1 \wedge w = b]} \\ &= \frac{1}{1 + \Phi(b)}, \end{aligned}$$

where

$$\begin{aligned} \Phi(b) &\triangleq \frac{\Pr[u = 1 \wedge w = b]}{\Pr[u = 0 \wedge w = b]} \\ &= \frac{\Pr[w = b | u = 1] \Pr[u = 1]}{\Pr[w = b | u = 0] \Pr[u = 0]} \end{aligned}$$

is decreasing in b by (38). \square

We will also need a stronger, *conditional*, version of Lemma 7.4 whose proof is essentially identical (and omitted).

Claim 7.6. *Suppose $S \subset \mathcal{T}_z$ and $u \neq z$ is a tree node such that \mathcal{T}_u is disjoint with S . Then*

$$\Pr[Z_S | z = 0, u = 1] \geq \Pr[Z_S | z = 1, u = 1]. \quad (39)$$

We will use Corollary 7.5 and Lemma 7.6 to prove (33) for the case $S \cap \mathcal{T}_z = \emptyset$.

Claim 7.7. *If S is disjoint with \mathcal{T}_z then (33) holds.*

Proof. Suppose S is disjoint with \mathcal{T}_z , and let r be the root of the tree. We will use induction on the tree to prove the following: for each $c \in \{0, 1\}$,

$$\Pr[Z_S | r = c, z = 0] \geq \Pr[Z_S | r = c, z = 1] \quad (40)$$

For the induction base, consider a tree of depth 2, consisting of the root r and the leaves. Then $z \notin S$ is a leaf, so Z_S is independent of $\pi(z)$ given $\pi(r)$, so (40) holds with equality.

For the induction step, fix $c \in \{0, 1\}$. Let us set up the notation similarly to the proof of Claim 7.4. Consider children u_i of r such that the intersection $S_i \triangleq S \cap \mathcal{T}_{u_i}$ is non-empty. Let u_1, \dots, u_k be all such children. Assume $z \in \mathcal{T}_{u_i}$ for some i (else, Z_S is independent from $\pi(z)$ given $\pi(r)$, so (40) holds with equality); without loss of generality, assume this happens for $i = 1$. For brevity, for $a, b \in \{0, 1\}$ denote

$$f_i(a, b) \triangleq \Pr[Z_{S_i} | u_i = a, z = b]$$

$$\nu_i(a|b) \triangleq \Pr[u_i = a | r = c, z = b].$$

Note that $f_i(a, b)$ and $\nu_i(a|b)$ do not depend on b for $i > 1$.

Then for each $b \in \{0, 1\}$

$$\begin{aligned} & \Pr[Z_S | r = c, z = b] \\ &= \sum_{a_i \in \{0, 1\}, i \geq 1} \prod_{i \geq 1} f_i(a_i, b) \nu_i(a_i|b) \\ &= \Phi \times \sum_{a \in \{0, 1\}} f_1(a, b) \nu_1(a|b), \end{aligned}$$

where

$$\Phi \triangleq \sum_{a_i \in \{0, 1\}, i \geq 2} \prod_{i \geq 2} f_i(a_i, b) \nu_i(a_i|b)$$

does not depend on b . Therefore:

$$\begin{aligned} & \Pr[Z_S | r = c, z = 1] - \Pr[Z_S | r = c, z = 1] \\ &= \Phi \times \sum_{a \in \{0,1\}} [f_1(a, 0) \nu_1(a|0) - f_1(a, 1) \nu_1(a|1)] \quad (41) \\ &\geq \Phi \times \sum_{a \in \{0,1\}} f_1(a, 1) [\nu_1(a|0) - \nu_1(a|1)] \quad (42) \\ &\geq \Phi \times f_1(1, 1) \sum_{a \in \{0,1\}} [\nu_1(a|0) - \nu_1(a|1)] \quad (43) \\ &= 0. \quad (44) \end{aligned}$$

The above transitions hold for the following reasons:

(41 \rightarrow 42) By Induction Hypothesis, $f_1(a, 0) \geq f_1(a, 1)$

(42 \rightarrow 43) By Lemma 7.6 $f_1(0, 1) \geq f_1(1, 1)$, and moreover we have $\nu_1(0|0) \geq \nu_1(0|1)$ by Corollary 7.5.

(43 \rightarrow 44) Since $\nu_i(0|0) + \nu_i(1|0) = \nu_i(0|1) + \nu_i(1|1) = 1$

This completes the proof of the inductive step. \square

8. Further directions

This paper initiates the study of online learning to rank in metric spaces, focusing on the “clean” similarity model (conditional L-continuity). As discussed in Section 5, we conjecture that provable performance guarantees can be improved significantly. On the experimental side, future work will include evaluating the model on web search data, and designing sufficiently memory- and time-efficient implementations to allow experiments on real users. An interesting challenge in such an endeavor would be to come up with effective similarity measures. A natural next step would be to also exploit the similarity between search queries.

References

- Agrawal, R. (1995). The continuum-armed bandit problem. *SIAM J. Control and Optimization*, 33(6):1926–1951.
- Aldous, D. J. (1985). Exchangeability and related topics. In *École d’Été de Probabilités de Saint-Flour XIII*, pages 1–198.
- Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002a). Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256.
- Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. (2002b). The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77.
- Auer, P., Ortner, R., and Szepesvári, C. (2007). Improved Rates for the Stochastic Continuum-Armed Bandit Problem. In *Conf. on Learning Theory (COLT)*, pages 454–478.
- Bartal, Y. (1996). Probabilistic approximations of metric spaces and its algorithmic applications. In *IEEE Symp. on Foundations of Computer Science (FOCS)*.
- Bubeck, S., Munos, R., Stoltz, G., and Szepesvári, C. (2008). Online Optimization in X-Armed Bandits. In *Advances in Neural Information Processing Systems (NIPS)*, pages 201–208.
- Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G. (2005). Learning to rank using gradient descent. In *Intl. Conf. on Machine Learning (ICML)*, pages 89–96.
- Carbonell, J. and Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *ACM Intl. Conf. on Research and Development in Information Retrieval (SIGIR)*, pages 335–336.
- Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge Univ. Press.
- Chen, H. and Karger, D. R. (2006). Less is more: Probabilistic models for retrieving fewer relevant documents. In *ACM Intl. Conf. on Research and Development in Information Retrieval (SIGIR)*, pages 429–436.
- Chu, W. and Ghahramani, Z. (2005). Gaussian processes for ordinal regression. *J. of Machine Learning Research*, 6:1019–1041.
- Fakcharoenphol, J., Rao, S., and Talwar, K. (2004). A tight bound on approximating arbitrary metrics by tree metrics. *J. of Computer and System Sciences*, 69(3):485–497.
- Gupta, A., Krauthgamer, R., and Lee, J. R. (2003). Bounded geometries, fractals, and low-distortion embeddings. In *IEEE Symp. on Foundations of Computer Science (FOCS)*.
- Hazan, E. and Megiddo, N. (2007). Online Learning with Prior Knowledge. In *Conf. on Learning Theory (COLT)*, pages 499–513.
- Kleinberg, R. (2004). Nearly tight bounds for the continuum-armed bandit problem. In *Advances in Neural Information Processing Systems (NIPS)*, pages 697–704.
- Kleinberg, R., Niculescu-Mizil, A., and Sharma, Y. (2008a). Regret bounds for sleeping experts and bandits. In *Conf. on Learning Theory (COLT)*, pages 425–436.

- Kleinberg, R. and Slivkins, A. (2010). Sharp Dichotomies for Regret Minimization in Metric Spaces. In *ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 827–846.
- Kleinberg, R., Slivkins, A., and Upfal, E. (2008b). Multi-Armed Bandits in Metric Spaces. In *ACM Symp. on Theory of Computing (STOC)*, pages 681–690.
- Pandey, S., Agarwal, D., Chakrabarti, D., and Josifovski, V. (2007). Bandits for Taxonomies: A Model-based Approach. In *SIAM Intl. Conf. on Data Mining (SDM)*.
- Radlinski, F., Kleinberg, R., and Joachims, T. (2008). Learning diverse rankings with multi-armed bandits. In *Intl. Conf. on Machine Learning (ICML)*, pages 784–791.
- Slivkins, A. (2009). Contextual Bandits with Similarity Information. <http://arxiv.org/abs/0907.3986>.
- Streeter, M. and Golovin, D. (2009). An online algorithm for maximizing submodular functions. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1577–1584.
- Taylor, M., Guiver, J., Robertson, S., and Minka, T. (2008). Softrank: optimizing non-smooth rank metrics. In *ACM Intl. Conf. on Web Search and Data Mining (WSDM)*, pages 77–86.